EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	3	"10055691".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 14:53
L2	2	"7024660".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 14:54
L3	2	"7085670".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 14:55
L4	2	"6311149".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 14:57
L5	0	"12242".ap.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 14:58
L6	0	"12242".an.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 14:58
L7	8	"312242".ap.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 14:58
L8	1	(Deploying and Creating).ti. and Kodosky.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 15:00

EAST Search History

L9	1	(Deploying and Creating).ti. and (Kodosky and Shah).in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 15:09
L10	9	@ad="20010814" and (Kodosky and Shah).in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/07/07 15:27
L11	0	(debug$4 and deploy$5).clm. and ((hardware near2 programmable) with execut$4).clm. and ((step iteration repeat$4) same (convert$4 and configur$4 and (user with debug$4))).clm.	US-PGPUB	OR	OFF	2007/07/07 15:30
L12	4	(debug$4 and deploy$5).clm. and ((hardware near2 programmable) with execut$4).clm.	US-PGPUB	OR	OFF	2007/07/07 15:30
L13	0	((step iteration repeat$4) same (convert$4 and configur$4 and (user with debug$4))).clm.	US-PGPUB	OR	OFF	2007/07/07 15:31
L14	4	12 and (hardware with programmable).clm. and (execut$4 same hardware).clm.	US-PGPUB	OR	OFF	2007/07/07 15:31
L15	1	14 and (convert$4 and configur$4 and (user with debug$4)).clm.	US-PGPUB	OR	OFF	2007/07/07 15:32

Terms used:
emulation and **converting hardware executi[illegible]** ,and **user debug session** and **iterative**

Found **156,549** of **205,978**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown   Relevance scale

1 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research CASCON '97**
Publisher: IBM Press
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

2 GPGPU: general purpose computation on graphics hardware
David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(63.03 MB) Additional Information: full citation, abstract, citings

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

3 Classics in software engineering
January 1979 Divisible Book
Publisher: Yourdon Press
Full text available: pdf(22.45 MB) Additional Information: full citation, cited by, index terms

4 Level set and PDE methods for computer graphics
David Breen, Ron Fedkiw, Ken Museth, Stanley Osher, Guillermo Sapiro, Ross Whitaker

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(17.07 MB) Additional Information: full citation, abstract, citings

Level set methods, an important class of partial differential equation (PDE) methods, define dynamic surfaces implicitly as the level set (iso-surface) of a sampled, evolving nD function. The course begins with preparatory material that introduces the concept of using partial differential equations to solve problems in computer graphics, geometric modeling and computer vision. This will include the structure and behavior of several different types of differential equations, e.g. the level set eq ...

5 Real-time shading
Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(7.39 MB) Additional Information: full citation, abstract

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering passes. Today, almost every new computer comes with graphics hardware capable of interactively executing shaders of thousands to tens of thousands of instructions. This course has been redesigned to address today's real-time shading capabili ...

6 The elements of nature: interactive and realistic techniques
Oliver Deusen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, Jerry Tessendorf
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(17.65 MB) Additional Information: full citation, abstract

This updated course on simulating natural phenomena will cover the latest research and production techniques for simulating most of the elements of nature. The presenters will provide movie production, interactive simulation, and research perspectives on the difficult task of photorealistic modeling, rendering, and animation of natural phenomena. The course offers a nice balance of the latest interactive graphics hardware-based simulation techniques and the latest physics-based simulation techni ...

7 Writing efficient programs
Jon Louis Bentley
January 1982 Book
Publisher: Prentice-Hall, Inc.
Additional Information: full citation, abstract, references, cited by, index terms

The primary task of software engineers is the cost-effective development of maintainable and useful software. There are many secondary problems lurking in that definition. One such problem arises from the term "useful": to be useful in the application at hand, software must often be efficient (that is, use little time or space). The problem we will consider in this book is building efficient software systems.

There are a number of levels at which we may confront the problem of efficien ...

8 Session 2: Running the manual: an approach to high-assurance microkernel development
Philip Derrin, Kevin Elphinstone, Gerwin Klein, David Cock, Manuel M. T. Chakravarty

September 2006 **Proceedings of the 2006 ACM SIGPLAN workshop on Haskell Haskell '06**
Publisher: ACM Press
Full text available: pdf(193.36 KB) Additional Information: full citation, abstract, references, index terms

We propose a development methodology for designing and prototyping high assurance microkernels, and describe our application of it. The methodology is based on rapid prototyping and iterative refinement of the microkernel in a functional programming language. The prototype provides a precise semi-formal model, which is also combined with a machine simulator to form a reference implementation capable of executing real user-level software, to obtain accurate feedback on the suitability of the kern ...

Keywords: Haskell, Isabelle/HOL, executable specification, formalisation, monads, operating systems, rapid prototyping, verification

9 Proceedings of the SIGNUM conference on the programming environment for development of numerical software
March 1979 **ACM SIGNUM Newsletter**, Volume 14 Issue 1
Publisher: ACM Press
Full text available: pdf(5.02 MB) Additional Information: full citation

10 A relational debugging engine for the graphics pipeline
Nathaniel Duca, Krzysztof Niski, Jonathan Bilodeau, Matthew Bolitho, Yuan Chen, Jonathan Cohen
July 2005 **ACM Transactions on Graphics (TOG) , ACM SIGGRAPH 2005 Papers SIGGRAPH '05**, Volume 24 Issue 3
Publisher: ACM Press
Full text available: pdf(582.04 KB) mov(24:11 MIN) Additional Information: full citation, abstract, references, index terms

We present a new, unified approach to debugging graphics software. We propose a representation of all graphics state over the course of program execution as a relational database, and produce a query-based framework for extracting, manipulating, and visualizing data from all stages of the graphics pipeline. Using an SQL-based query language, the programmer can establish functional relationships among all the data, linking OpenGL state to primitives to vertices to fragments to pixels. Based on th ...

Keywords: SIMD, SQL, debugging, graphics hardware, graphics pipeline, relational algebra, streaming, visualization

11 Charles W. Bachman interview: September 25-26, 2004; Tucson, Arizona
Thomas Haigh
January 2006 **ACM Oral History interviews**
Publisher: ACM Press
Full text available: pdf(761.66 KB) Additional Information: full citation, abstract

Charles W. Bachman reviews his career. Born during 1924 in Kansas, Bachman attended high school in East Lansing, Michigan before joining the Army Anti Aircraft Artillery Corp, with which he spent two years in the Southwest Pacific Theater, during World War II. After his discharge from the military, Bachman earned a B.Sc. in Mechanical Engineering in 1948, followed immediately by an M.Sc. in the same discipline, from the University of Pennsylvania. On graduation, he went to work for Do ...

12 Final report of the GSPC state-of-the-art subcommittee
R. H. Ewald, R. Fryer
June 1978 **ACM SIGGRAPH Computer Graphics**, Volume 12 Issue 1-2
Publisher: ACM Press
Full text available: pdf(7.85 MB) Additional Information: full citation, abstract

This paper presents the final report of the ACM/SIGGRAPH Graphics Standards Planning Committee (GSPC) State-of-the-Art Subcommittee. This group's charter was to compare existing vector-oriented graphics packages to determine their similarities and differences. Eight graphics packages and the GSPC "Core System" were selected for review.

13 Self
David Ungar, Randall B. Smith
June 2007 **Proceedings of the third ACM SIGPLAN conference on History of programming languages HOPL III**
Publisher: ACM Press
Full text available: pdf(1.70 MB) Additional Information: full citation, abstract, references, index terms

The years 1985 through 1995 saw the birth and development of the language Self, starting from its design by the authors at Xerox PARC, through first implementations by Ungar and his graduate students at Stanford University, and then with a larger team formed when the authors joined Sun Microsystems Laboratories in 1991. Self was designed to help programmers become more productive and creative by giving them a simple, pure, and powerful language, an implementation that combined ease of use wit ...

Keywords: Self, adaptive optimization, cartoon animation, dynamic language, dynamic optimization, exploratory programming, history of programming languages, morphic, object-oriented language, programming environment, prototype-based programming language, virtual machine

14 IS '97: model curriculum and guidelines for undergraduate degree programs in information systems
Gordon B. Davis, John T. Gorgone, J. Daniel Couger, David L. Feinstein, Herbert E. Longenecker
December 1996 **ACM SIGMIS Database , Guidelines for undergraduate degree programs on Model curriculum and guidelines for undergraduate degree programs in information systems IS '97**, Volume 28 Issue 1
Publisher: ACM Press
Full text available: pdf(7.24 MB) Additional Information: full citation, citings

15 Assembly instruction level reverse execution for debugging
Tankut Akgul, Vincent J. Mooney III
April 2004 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 13 Issue 2
Publisher: ACM Press
Full text available: pdf(1.18 MB) Additional Information: full citation, abstract, references, index terms

Assembly instruction level reverse execution provides a programmer with the ability to return a program to a previous state in its execution history via execution of a "reverse program." The ability to execute a program in reverse is advantageous for shortening software development time. Conventional techniques for recovering a state rely on saving the state into a record before the state is destroyed. However, state-saving causes significant memory and time overheads during forward execution.Th ...

Keywords: Debugging, reverse code generation, reverse execution

16 A history of Haskell: being lazy with class
Paul Hudak, John Hughes, Simon Peyton Jones, Philip Wadler
June 2007 **Proceedings of the third ACM SIGPLAN conference on History of programming languages HOPL III**
Publisher: ACM Press
Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, index terms

This paper describes the history of Haskell, including its genesis and principles, technical contributions, implementations and tools, and applications and impact.

17 Using general-purpose programming languages for FPGA design
Brad L. Hutchings, Brent E. Nelson
June 2000 **Proceedings of the 37th conference on Design automation DAC '00**
Publisher: ACM Press
Full text available: pdf(287.38 KB) Additional Information: full citation, abstract, references, citings, index terms

General-purpose programming languages (GPL) are effective vehicles for FPGA design because they are easy to use, extensible, widely available, and can be used to describe both the hardware and software aspects of a design. The strengths of the GPL approach to circuit design have been demonstrated by JHDL, a Java-based circuit design environment used to develop several large FPGA-based applications at several institutions. Major strengths of the JHDL environment include a common run-time for ...

18 A structural view of the Cedar programming environment
Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann
August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4
Publisher: ACM Press
Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

19 Software reuse
Charles W. Krueger
June 1992 **ACM Computing Surveys (CSUR)**, Volume 24 Issue 2
Publisher: ACM Press
Full text available: pdf(4.96 MB) Additional Information: full citation, abstract, references, citings, index terms

Software reuse is the process of creating software systems from existing software rather than building software systems from scratch. This simple yet powerful vision was introduced in 1968. Software reuse has, however, failed to become a standard software engineering practice. In an attempt to understand why, researchers have renewed their interest in software reuse and in the obstacles to implementing it. This paper surveys the different approaches to software reuse found in the ...

Keywords: abstraction, cognitive distance, software reuse

20 Extending ACID semantics to the file system

Charles P. Wright, Richard Spillane, Gopalan Sivathanu, Erez Zadok
June 2007 **ACM Transactions on Storage (TOS)**, Volume 3 Issue 2
Publisher: ACM Press
Full text available: pdf(783.03 KB) Additional Information: full citation, abstract, references, index terms

An organization's data is often its most valuable asset, but today's file systems provide few facilities to ensure its safety. Databases, on the other hand, have long provided transactions. Transactions are useful because they provide atomicity, consistency, isolation, and durability (ACID). Many applications could make use of these semantics, but databases have a wide variety of nonstandard interfaces. For example, applications like mail servers currently perform elaborate error handling to ...

Keywords: File system transactions, databases, file systems, ptrace monitors, recoverable memory

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

PORTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

debugged program <and> deploy emulation <and> convertinc SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Sat

Terms used:
debugged program and **deploy emulation** and **converting configurating hardware execution** and **user debu**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Se
Try this search in Th

Results 1 - 20 of 200

Best 200 shown

Result page: **1** 2 3 4 5 6 7 8 9 10 next

1 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on research CASCON '97**
Publisher: IBM Press
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index term

Understanding distributed applications is a tedious and difficult task. Visualizations based on pro often used to obtain a better understanding of the execution of the application. The visualization event tracer developed at the University of Waterloo. However, these diagrams are often very co provide the user with the desired overview of the application. In our experience, such tools displ of non-trivial commun ...

2 GPGPU: general purpose computation on graphics hardware
David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aa
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(63.03 MB) Additional Information: full citation, abstract, citings

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely processor. The latest graphics architectures provide tremendous memory bandwidth and compu fully programmable vertex and pixel processing units that support vector operations up to full IE precision. High level languages have emerged for graphics hardware, making this computational Architecturally, GPUs are highly parallel s ...

3 IS '97: model curriculum and guidelines for undergraduate degree programs in information
Gordon B. Davis, John T. Gorgone, J. Daniel Couger, David L. Feinstein, Herbert E. Longenecker
December 1996 **ACM SIGMIS Database , Guidelines for undergraduate degree programs on guidelines for undergraduate degree programs in information systems IS '**
Publisher: ACM Press
Full text available: pdf(7.24 MB) Additional Information: full citation, citings

4 Classics in software engineering
January 1979 Divisible Book

Publisher: Yourdon Press
Full text available: pdf(22.45 MB) Additional Information: full citation, cited by, index terms

5 Extending ACID semantics to the file system
Charles P. Wright, Richard Spillane, Gopalan Sivathanu, Erez Zadok
June 2007 **ACM Transactions on Storage (TOS)**, Volume 3 Issue 2
Publisher: ACM Press
Full text available: pdf(783.03 KB) Additional Information: full citation, abstract, references, index term

An organization's data is often its most valuable asset, but today's file systems provide few facili Databases, on the other hand, have long provided transactions. Transactions are useful because consistency, isolation, and durability (ACID). Many applications could make use of these semant wide variety of nonstandard interfaces. For example, applications like mail servers currently perl handling to ...

Keywords: File system transactions, databases, file systems, ptrace monitors, recoverable mer

6 Using general-purpose programming languages for FPGA design
Brad L. Hutchings, Brent E. Nelson
June 2000 **Proceedings of the 37th conference on Design automation DAC '00**
Publisher: ACM Press
Full text available: pdf(287.38 KB) Additional Information: full citation, abstract, references, citings, ind

General-purpose programming languages (GPL) are effective vehicles for FPGA design because t extensible, widely available, and can be used to describe both the hardware and software aspecl strengths of the GPL approach to circuit design have been demonstrated by JHDL, a Java-based environment used to develop several large FPGA-based applications at several institutions. Majo environment include a common run-time for ...

7 Computing curricula 2001
September 2001 **Journal on Educational Resources in Computing (JERIC)**
Publisher: ACM Press
Full text available: pdf(613.63 KB) html(2.78 KB) Additional Information: full citation, references, citings, index terms

8 Level set and PDE methods for computer graphics
David Breen, Ron Fedkiw, Ken Museth, Stanley Osher, Guillermo Sapiro, Ross Whitaker
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(17.07 MB) Additional Information: full citation, abstract, citings

Level set methods, an important class of partial differential equation (PDE) methods, define dyn. as the level set (iso-surface) of a sampled, evolving nD function. The course begins with prepare introduces the concept of using partial differential equations to solve problems in computer grap and computer vision. This will include the structure and behavior of several different types of dif the level set eq ...

9 Real-time shading
Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press

Full text available: pdf(7.39 MB) Additional Information: full citation, abstract

Real-time procedural shading was once seen as a distant dream. When the first version of this c
years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combinin
hundreds of rendering passes. Today, almost every new computer comes with graphics hardwar
executing shaders of thousands to tens of thousands of instructions. This course has been redes
real-time shading capabili ...

10 Practitioners report: The parks PDA: a handheld device for theme park guests in squeak
Yoshiki Ohshima, John Maloney, Andy Ogden
October 2003 **Companion of the 18th annual ACM SIGPLAN conference on Object-oriented systems, languages, and applications OOPSLA '03**
Publisher: ACM Press
Full text available: pdf(488.82 KB) Additional Information: full citation, abstract, references, index term

The Parks PDA is a lightweight, handheld device for theme park guests that functions as a comb
and digital camera. Together with a small team of artists and designers, we created a prototype
for a three hour guest experience, including a camera interface, a hyper-linked guide book, thre
spotters guide, a cross-referenced map, animated movies with lip-synched sound, a ride reserva
Over 800 visitors to Disney's An ...

Keywords: PDA, development environment, end-user software, handheld device, multimedia d
software development

11 Writing efficient programs
Jon Louis Bentley
January 1982 Book
Publisher: Prentice-Hall, Inc.
Additional Information: full citation, abstract, references, cited by, index terms

The primary task of software engineers is the cost-effective development of maintainable and us
many secondary problems lurking in that definition. One such problem arises from the term "use
application at hand, software must often be efficient (that is, use little time or space). The probl
this book is building efficient software systems.

There are a number of levels at which we may confront the problem of efficien ...

12 Special issue: AI in engineering
D. Sriram, R. Joobbani
April 1985 **ACM SIGART Bulletin**, Issue 92
Publisher: ACM Press
Full text available: pdf(8.79 MB) Additional Information: full citation, abstract

The papers in this special issue were compiled from responses to the announcement in the July
newsletter and notices posted over the ARPAnet. The interest being shown in this area is reflecte
received from over six countries. About half the papers were received over the computer networ

13 A structural view of the Cedar programming environment
Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann
August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume
Publisher: ACM Press
Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, ind

This paper presents an overview of the Cedar programming environment, focusing on its overall
major components of Cedar and the way they are organized. Cedar supports the development o
single programming language, also called Cedar. Its primary purpose is to increase the producti

whose activities include experimental programming and the development of prototype software performance personal computer. T ...

14 A history of Haskell: being lazy with class
Paul Hudak, John Hughes, Simon Peyton Jones, Philip Wadler
June 2007 **Proceedings of the third ACM SIGPLAN conference on History of programmin**
Publisher: ACM Press
Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, index term

This paper describes the history of Haskell, including its genesis and principles, technical contrib and tools, and applications and impact.

15 Self
David Ungar, Randall B. Smith
June 2007 **Proceedings of the third ACM SIGPLAN conference on History of programmin**
Publisher: ACM Press
Full text available: pdf(1.70 MB) Additional Information: full citation, abstract, references, index term

The years 1985 through 1995 saw the birth and development of the language Self, starting from authors at Xerox PARC, through first implementations by Ungar and his graduate students at St then with a larger team formed when the authors joined Sun Microsystems Laboratories in 1991 help programmers become more productive and creative by giving them a simple, pure, and po implementation that combined ease of use wit ...

Keywords: Self, adaptive optimization, cartoon animation, dynamic language, dynamic optimiz programming, history of programming languages, morphic, object-oriented language, programn prototype-based programming language, virtual machine

16 Draft report on requirements for a common prototyping system
R. P. Gabriel
March 1989 **ACM SIGPLAN Notices**, Volume 24 Issue 3
Publisher: ACM Press
Full text available: pdf(4.76 MB) Additional Information: full citation, citings, index terms

17 Charles W. Bachman interview: September 25-26, 2004; Tucson, Arizona
Thomas Haigh
January 2006 **ACM Oral History interviews**
Publisher: ACM Press
Full text available: pdf(761.66 KB) Additional Information: full citation, abstract

Charles W. Bachman reviews his career. Born during 1924 in Kansas, Bachman attended high sc Michigan before joining the Army Anti Aircraft Artillery Corp, with which he spent two years in th Theater, during World War II. After his discharge from the military, Bachman earned a B.Sc. in I 1948, followed immediately by an M.Sc. in the same discipline, from the University of Pennsylva went to work for Do ...

18 Sensor networks and performance analysis: Java™ on the bare metal of wireless sensor c Java virtual machine
Doug Simon, Cristina Cifuentes, Dave Cleal, John Daniels, Derek White
June 2006 **Proceedings of the second international conference on Virtual execution env**

Publisher: ACM Press
Full text available: pdf(999.55 KB) Additional Information: full citation, abstract, references, cited by, in

The Squawk virtual machine is a small Java™ virtual machine (VM) written mostly in Java that r system on a wireless sensor platform. Squawk translates standard class file into an internal pre-independent format that is compact and allows for efficient execution of bytecodes that have be only memory. In addition, Squawk implements an application isolation mechanism whereby app as object and are therefore ...

Keywords: IEEE 802.15.4, Java virtual machine, Sun SPOT, embedded systems, wireless sensc

19 TSOtool: A Program for Verifying Memory Systems Using the Memory Consistency Model
Sudheendra Hangal, Durgam Vahia, Chaiyasit Manovit, Juin-Yeu Joseph Lu
March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual symposium on Computer architecture ISCA '04**, Volume 32 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(206.30 KB) Additional Information: full citation, abstract, citings

In this paper, we describe TSOtool, a program to check thebehavior of the memory subsystem ir memorymultiprocessor. TSOtool runs pseudo-randomly generatedprograms with data races on a theTotal Store Order (TSO) memory consistency model; it thenchecks the results of the progran TSOspecification. Such analysis can expose subtle memory errorslike data corruption, atomicity illegalinstruction ordering.While verifying TSO compliance comp ...

Keywords: Memory consistency models, Multiprocessor verification, Sequential Consistency, To

20 Compiler construction: an advanced course
F. L. Bauer, F. L. De Remer, M. Griffiths, U. Hill, J. J. Horning, C. H. A. Koster, W. M. McKeeman, P G. Goos, J. Hartmanis
January 1974 Book
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(65.62 MB) Additional Information: full citation, abstract, references, cited by

The Advanced Course took place from March 4 to 15, 1974 and was organized by the Mathemati Technical University of Munich and the Leibniz Computing Center of the Bavarian Academy of Sc with the European Communities, sponsored by the Ministry for Research and Technology of the I Germany and by the European Research Office, London.

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next